新集群部署：

后台还没布上，客户端发给后台一个地址，后台仍然返回客户端一个旧地址，所以新流媒体还待部署。。。

<http://1.119.138.66:10080/index.html> 心神学堂

<http://1.119.138.66:10082/dataLayout> 智慧教室

18401605753 13241283459

更改xinshen-sm-01的root密码为XS-root-123

sudo passwd root更改root密码命令。

Hadoop生态集群搭建

前提安装JDK，apt-get update

JDK安装目录/root/jdk1.8.0\_141

apt-get install default-jdk等，这里已经安装好了1.8版本的jdk。Java -version

查询java安装路径: update-alternatives --display java

1.12没找到，重新安装（出现问题apt-get -f install，未解决）

1.7 /usr/lib/jvm/java-8-oracle/jre/bin/java

1.8 /usr/lib/jvm/java-8-oracle/jre/bin/java

1.11 /usr/lib/jvm/java-8-oracle/jre/bin/java

设置SSH无密码登录

安装SSH，安装rsync

每一台都生成一个公钥，写在一起

生成密钥：ssh-keygen -t dsa -P '' -f ~/.ssh/id\_dsa

位置： /root/.ssh/id\_dsa. /root/.ssh/id\_dsa.pub.

将产生的公钥（key）放置到许可证文件中：cat ~/.ssh/id\_dsa.pub >> ~/.ssh/authorized\_keys

下载安装Hadoop

wget [http://mirror.bit.edu.cn/apache/hadoop/common/hadoop-3.1.1/hadoop-3.1.1.tar.gz && tar -zxvf hadoop-3.1.1.tar.gz](http://mirror.bit.edu.cn/apache/hadoop/common/hadoop-3.1.1/hadoop-3.1.1.tar.gz%20&&%20tar%20-zxvf%20hadoop-3.1.1.tar.gz)

移动目录：mv hadoop-3.1.1 /usr/local/hadoop

设置hadoop环境变量，在/.bashrc中：

vim ~/.bashrc 并添加如下内容：

export JAVA\_HOME=/usr/lib/jvm/java-8-oracle

export HADOOP\_HOME=/usr/local/hadoop

export PATH=$PATH:$HADOOP\_HOME/bin

export PATH=$PATH:$HADOOP\_HOME/sbin

export HADOOP\_MAPRED\_HOME=$HADOOP\_HOME

export HADOOP\_COMMON\_HOME=$HADOOP\_HOME

export HADOOP\_HDFS\_HOME=$HADOOP\_HOME

export HADOOP\_YARN\_HOME=$HADOOP\_HOME

export HADOOP\_COMMON\_LIB\_NATIVE\_DIR=$HADOOP\_HOME/lib/native

export HADOOP\_OPTS="-Djava.library.path=$HADOOP\_HOME/lib"

export JAVA\_LIBRARY\_PATH=$HADOOP\_HOME/lib/native:$JAVA\_LIBRARY\_PATH

source ~/.bashrc

修改hadoop配置文件，五个，

Hadoop-env.sh（hadoop的配置文件，必须设置java的安装路径）

core-site.xml（在这儿设置HDFS的默认名称）

yarn-site/xml（含有Mapreduce2{yarn}的相关配置）

mapred-site.xml（设置监控Map和Reduce程序的jobTracker任务分配情况以及TaskTracker的任务运行情况）

hdfs-site.xml（用于设置HDFS分布式文件系统）

vim /usr/local/hadoop/etc/hadoop/hadoop-env.sh 并添加：

export JAVA\_HOME=/usr/lib/jvm/java-8-oracle

vim /usr/local/hadoop/etc/hadoop/core-site.xml

<property>

<name>fs.default.name</name>

<value>hdfs://localhost:9000</value>

</property>

vim /usr/local/hadoop/etc/hadoop/yarn-site.xml

<property>

<name>yarn.nodemanager.aux-services</name>

<value>mapreduce\_shuffle</value>

</property>

<property>

<name>yarn.nodemanager.aux-services.mapreduce.shuffle.class</name>

<value>org.apache.hadoop.mapred.ShuffleHandler</value>

</property>

复制模板文件(我在这里做了个备份)，由mapred-site.xml.template至mapred-site.xml

cp /usr/local/hadoop/etc/hadoop/mapred-site.xml /usr/local/hadoop/etc/hadoop/mapred-site.xml.template

设置mapreduce的框架为yarn：

vim /usr/local/hadoop/etc/hadoop/mapred-site.xml

<property>

<name>mapreduce.framework.name</name>

<value>yarn</value>

</property>

设置namenode和datanode的数据存储目录：

vim /usr/local/hadoop/etc/hadoop//hdfs-site.xml

<property>

<name>dfs.replication</name>

<value>3</value>

</property>

<peoperty>

<name>dfs.namenode.name.dir</name>

<value>file:/usr/local/hadoop/hadoop\_data/hdfs/namenode</value>

</property>

<peoperty>

<name>dfs.datanode.data.dir</name>

<value>file:/usr/local/hadoop/hadoop\_data/hdfs/datanode</value>

</property>

接着创建刚刚在配置文件中设置的目录，哈哈，并将hadoop目录的所有者更改为xinshen（因为linux是多人多任务的操作系统，所有目录或文件都具有所有者，chown可以将目录或文件所有者更改）？

sudo mkdir -p /usr/local/hadoop/hadoop\_data/hdfs/namenode

sudo mkdir -p /usr/local/hadoop/hadoop\_data/hdfs/datanode

sudo chown xinshen:xinshen -R /usr/local/Hadoop

HDFS进行格式化

hadoop namenode -format

启动Hadoop

1. 启动HDFS:start-dfs.sh
2. 启动YARN：start-yarn.sh

3.也可以用start-all.sh

4.jps(集群未启动，单节点)

1. 修改host文件

vim /etc/hosts

192.168.1.7 master-hadoop

192.168.1.8 slave2-hadoop

192.168.1.11 slave3-hadoop

1. 配置slaves

关键技术解释：

Pig：基于Hadoop的大规模数据分析平台，pig为复杂的海量数据并行计算提供简单的操作和编程接口。

Hdfs：基于hadoop的工具，提供完整的SQL查询，可将sql语句转换为Mapreduce任务进行运行。

Mapreduce：一种编程模型，用于大规模数据的的并行计算

Yarn

Shuffle

Hive

Docker+Hadoop集群搭建

<https://blog.csdn.net/csj941227/article/details/80025066>

1. 基于Ubuntu镜像，创建容器，并安装JDK，并配好java环境

进入容器：docker exec -it 61ca /bin/bash 安装jdk12.1

打包镜像：docker commit -m “…”

下载并解压hive，hadoop，hbase，spark，zookeeper，scala

更改环境变量

Hadoop尝试是否安装成功

安装ssh服务 openssh-server并启动服务service ssh start

生成公钥：ssh-keygen -t rsa

把生成的公钥放入本机认证的密钥中：cat /root/.ssh/id\_rsa.pub >>/root/.ssh/authorized\_keys

Ssh测试本机是否可以免密登录：ssh localhost

1. 配置hadoop
   1. 切入配置目录：cd /root/hadoop-3.1.2/etc/Hadoop/
   2. 修改core-site.xml

<configuration>

<property>

<name>fs.defaultFS</name>

<value>hdfs://localhost:9000</value>

</property>

<property>

<name>io.file.buffer.size</name>

<value>4096</value>

</property>

<property>

<name>hadoop.tmp.dir</name>

<value>/tmp</value>

</property>

#用户root可以代理所有主机上的所有组的用户

<property>

<name>hadoop.proxyuser.root.hosts</name>

<value>\*</value>

</property>

<property>

<name>hadoop.proxyuser.root.groups</name>

<value>\*</value>

</property>

</configuration>

* 1. 修改hadoop-env.sh

export JAVA\_HOME=/usr/lib/jvm/jdk-12.0.1

export JAVA\_HOME=${JAVA\_HOME}

* 1. 修改hdfs-site.xml

<configuration>

<property>

<name>dfs.replication</name>

<value>2</value>

</property>

<property>

<name>dfs.namenode.name.dir</name>

<value>/root/hadoop-3.1.2/hdfs/name</value>

</property>

<property>

<name>dfs.datanode.data.dir</name>

<value>/root/hadoop-3.1.2/hdfs/data</value>

</property>

<property>

<name>dfs.permissions</name>

<value>false</value>

</property>

</configuration>

* 1. 修改mapred-site.xml

<property>

<name>mapreduce.framework.name</name>

<value>yarn</value>

</property>

* 1. 修改slave文件，记录hadoop的datanode地址

Vim slaves 输入localhost

* 1. 修改yarn-site.xml

<property>

<description>The hostname of the RM.</description>

<name>yarn.resourcemanager.hostname</name>

<value>master</value>

</property>

<property>

<name>yarn.nodemanager.aux-services</name>

<value>mapreduce\_shuffle</value>

</property>

Aux-services实现了**ServiceStateChangeListener**一个**EventHandler**接口，能够处理**AuxServicesEventType**事件。

1. 配置spark
   1. 进入spark配置目录：cd /root/spark-2.4.3-bin-hadoop2.7/conf
   2. 修改slaves文件：vim slaves并且加入localhost
   3. 修改spark-defaults.conf：

spark.master spark://localhost:7077

spark.eventLog.enabled true

spark.eventLog.dir hdfs://localhost:9000/historyserverforSpark

spark.yarn.historyServer.address localhost:18080

spark.history.fs.logDirectory hdfs://localhost:9000/historyserverforSpark

spark.serializer org.apache.spark.serializer.KryoSerializer

spark.driver.memory 4g

spark.executor.extraJavaOptions -XX:+PrintGCDetails -Dkey=value -Dnumbers="one two three"

* 1. 修改spark-env.sh

export JAVA\_HOME=/usr/lib/jvm/jdk-12.0.1

export SCALA\_HOME=/root/scala-2.11.11

export HADOOP\_HOME=/root/hadoop-3.1.2

export HADOOP\_CONF\_DIR=/root/hadoop-3.1.2/etc/hadoop

export SPARK\_MASTER\_IP=localhost

export SPARK\_WORKER\_MEMORY=4g

export SPARK\_EXECUTOR\_MEMORY=4g

export SPARK\_DRIVER\_MEMORY=4g

export SPARK\_WORKER\_CORES=2

export SPARK\_MEM=4G

1. 保存容器状态

docker commit 61ca2bef0dcc 625zyw/ubuntu\_bigdata

1. 使用镜像创建容器

Docker run -itd 镜像名 /bin/bash

1. 查看每个容器的ip

Ifconfig

三个ip（计划这样安排）：

172.18.0.3 master 0f40

172.18.0.4 slave1 b778

172.18.0.5 slave2 0718

按照上面修改每个容器的hosts文件，增加如下：

172.18.0.5 07188c442353

172.18.0.4 b7784c7973dc

172.18.0.3 0f40487c3855

1. 对每个结点启动ssh服务

Service ssh start

1. 对每个结点，都将其公钥拷贝到其他节点的认证密钥中

Ssh-copy-id：是将本机的公钥复制到远程主机的authorized\_keys文件中。

使用参数-i,当没有值传递时，或者identify.pub文件不可访问的时候，ssh-copy-id将显示错误信息。

ssh-copy-id -i ~/.ssh/id\_rsa.pub 172.18.0.5

对每个节点，ssh其他节点验证是否能够免密登录

Ssh ip或者hostname，已验证都成功。(hostname 还要输入yes，在这里我们统一用IP地址)

1. Hadoop集群配置

对其中一个节点，hadoop配置文件。

* 1. 修改core-site.xml

将localhost（master）改为master的主机名

* 1. 修改slave文件

Vim slaves 把slave的hostname写进去

* 1. 修改yarn-site.xml

把master改成0f40487c3855

* 1. 将上述配置覆盖到其他节点

参数-r（递归复制整个目录）

scp -r /root/hadoop-3.1.2/etc/hadoop [root@b7784c7973dc:/root/hadoop-3.1.2/etc/](mailto:root@b7784c7973dc:/root/hadoop-3.1.2/etc/)

* 1. 执行source。。。使环境变量生效

source ~/.bashrc

* 1. 在master上初始化namenode

hadoop namenode -format

* 1. 切换到hadoop/sbin下启动hadoop集群：start-all.sh。。。jps查看hadoop集群搭建成功！

问题一（已解决）

执行hdfs和yarn的脚本时出错

解决：在start-dfs.sh，stop-dfs.sh中，开头加上：

HDFS\_DATANODE\_USER=root

HADOOP\_SECURE\_DN\_USER=hdfs

HDFS\_NAMENODE\_USER=root

HDFS\_SECONDARYNAMENODE\_USER=root

（有datanode版）

HDFS\_NAMENODE\_USER=root

HDFS\_DATANODE\_USER=root

HDFS\_SECONDARYNAMENODE\_USER=root

YARN\_RESOURCEMANAGER\_USER=root

YARN\_NODEMANAGER\_USER=root

在start-yarn.sh,stop-yarn.sh中，开头加上：

YARN\_RESOURCEMANAGER\_USER=root

HADOOP\_SECURE\_DN\_USER=yarn

YARN\_NODEMANAGER\_USER=root

问题二（已解决）

slave节点未启动datanode,需要在每一台上都执行./start-all.sh

问题三（已解决）

Nodemanager进程一起来就挂掉，并且slave节点并不能自己启动datanode

查看nodemanager日志，找到错误有：

2019-06-26 03:06:14,724 WARN org.eclipse.jetty.webapp.WebAppContext: Failed startup of context o.e.j.w.WebAppContext@4c447c09{/,file:///tmp/jetty-0.0.0.0-8042-node-\_-any-17412868841284425825.dir/webapp/,UNAVAILABLE}{/node}

com.google.inject.ProvisionException: Unable to provision, see the following errors:

1. Error injecting constructor, java.lang.NoClassDefFoundError: javax/activation/DataSource

at com.sun.jersey.spi.inject.Errors.processWithErrors(Errors.java:193)

Caused by: java.lang.NoClassDefFoundError: javax/activation/DataSource

2019-06-26 03:06:14,782 ERROR org.apache.hadoop.yarn.server.nodemanager.webapp.WebServer: NMWebapps failed to start.

org.apache.hadoop.yarn.webapp.WebAppException: Error starting http server

网上找了两天的教程，并没有什么卵用，最后还是换回jdk8，完美解决

* 1. 使用hdfs命令创建一个目录以及文件测试hdfs功能

目录：hadoop fs -mkdir /zhuyuanwei

文件：vim test.txt

文件传入HDFS：hadoop fs -put test.txt /zhuyuanwei

* 1. 运行wordcount测试

hadoop jar hadoop-mapreduce-examples-3.1.2.jar wordcount /zhuyuanwei/test.txt /out

出问题，在mapred-site.sh中加

<property>

<name>yarn.app.mapreduce.am.env</name>

<value>HADOOP\_MAPRED\_HOME=${HADOOP\_HOME}</value>

</property>

<property>

<name>mapreduce.map.env</name>

<value>HADOOP\_MAPRED\_HOME=${HADOOP\_HOME}</value>

</property>

<property>

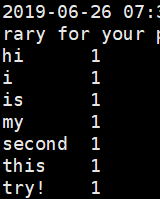
<name>mapreduce.reduce.env</name>

<value>HADOOP\_MAPRED\_HOME=${HADOOP\_HOME}</value>

</property>

hadoop fs -lsr /

hadoop fs -text /out/part-r-00000



1. Spark集群配置
   1. 修改slaves
   2. 修改spark-defaults.conf文件

全体换成Master

* 1. 修改spark-env.sh文件，并且传给其他节点
  2. 创建spark的history目录
  3. 启动spark集群：./start-all.sh
  4. 切换到bin目录，进行scala环境测试，进入spark-shell：./spark-shell

再次出错：Java HotSpot(TM) 64-Bit Server VM warning: INFO: os::commit\_memory(0x0000000718780000, 35086336, 0) failed; error='Cannot allocate memory' (errno=12)内存不够（还未解决）

解决方案1.把容器打包成镜像提交到docker hub，然后换实体机拉取镜像。在重新操作。

1. 集群环境的使用

重启hadoop和spark脚本编写：

HADOOP\_HOME=”/root/ hadoop-3.1.2”

SPARK\_HOME=” /root/spark-2.4.3-bin-hadoop2.7”

cd ${HADOOP\_HOME}/sbin && ./stop-all.sh

cd ${HADOOP\_HOME} && rm -r logs && mkdir logs

cd ${HADOOP\_HOME}/hdfs/name/ && rm -r current

cd ${HADOOP\_HOME}/hdfs/data/ && rm -r current

cd ${SPARK\_HOME}/sbin && ./stop-all.sh